

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2001-216168
(P2001-216168A)

(43) 公開日 平成13年8月10日(2001.8.10)

(51) Int.Cl.⁷

識別記号

F I
G 06 E 9/44

テ-マコ-ト(参考)

テ-マコ-ト(参考) 5B081

実施講義・実験講義の範囲 Q1 (合 12 頁)

(21) 出願番号 特願2000-28277(P2000-28277)

(71)出願人 000005821
松下電器産業株式会社
大阪府守口市守口町1006番地

(22)出願日 平成12年2月4日(2000.2.4)

(72) 発明者 中嶋 廣二
大阪府門真市大字門真1006番地 松下電器
産業株式会社

(72) 発明者 小谷 謙介
大阪府門真市大字門真1006番地 松下電器
産業株式会社

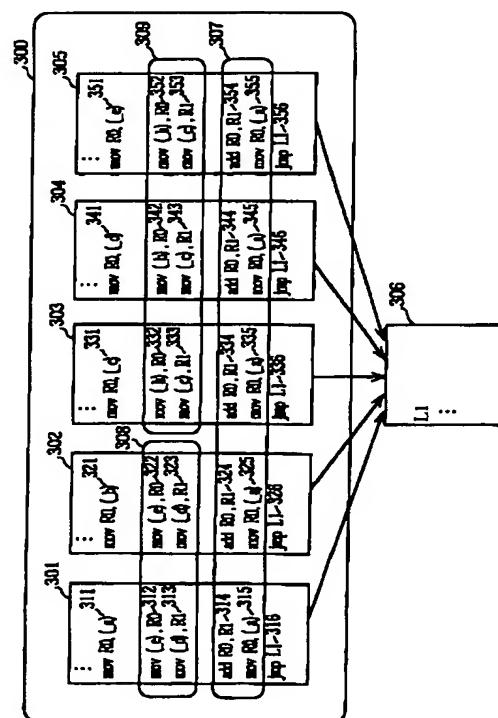
(74) 代理人 100090446
弁理士 中島 司朗 (外1名)

(54) 【発明の名称】 プログラム変換装置、プログラム変換方法及びプログラム記憶媒体

(57) 【要約】

【課題】 本発明は、テールマージ最適化後のプログラムにおいて実行される分岐命令数の増加を最小に抑えることにより、実行時間を最短にするプログラムを生成するプログラム変換装置を提供する。

【解決手段】 同一の基本ブロック 306 に無条件に分岐する複数の基本ブロック 301、302、303、304、305 を対象として、それらの後尾から先頭に向けて階層的に存在する共通命令列集合 307、308、309 を抽出する抽出手段と、抽出された各共通命令列集合において 1 の基本ブロックを命令列が含まれる基本ブロックの位置関係に応じて選択する選択手段と、選択された基本ブロックに含まれる命令列が共用されるようにプログラムを変換するコード変換手段とを備える。



を1づつ選ぶ組合せを全て列挙する組合せ列挙部と、列挙された各々の組合せに従って基本ブロックを選択して前記変換を行った場合に変更される分岐命令のコードサイズをあらかじめ算出する分岐命令サイズ算出部とを備え、算出されるコードサイズが最も小さい組合せに従って基本ブロックを選択することを特徴とする請求項1または2記載のプログラム変換装置。

【請求項7】 前記プログラム変換装置は、さらに、前記変換により削除される命令列および変更される分岐命令についてコードサイズの変化量を算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換手段に対して制御する制御手段を備えることを特徴とする請求項1または2記載のプログラム変換装置。

【請求項8】 複数の基本ブロックの後尾にある同一の命令列についてそのうちの1が共用されるようにプログラムを変換してコードサイズを削減するプログラム変換装置において用いられるプログラム変換方法であって、特定の基本ブロックに無条件に後続する他の複数の基本ブロックを対象として、それらの異なる基本ブロック内に存在する同一の命令列の集合であって、それらの基本ブロックの後尾から先頭に向けて階層的に存在する共通命令列集合を抽出する抽出手段と、抽出された各共通命令列集合において何れか1の命令列を含む基本ブロックを、他の共通命令列集合における基本ブロックとの位置関係に応じて選択する選択手段と、選択された基本ブロック中の命令列が他の基本ブロックにおいて共用されるようにプログラムを変換するコード変換手段とを備えることを特徴とするプログラム変換装置。

【請求項9】 前記抽出手段は、複数の基本ブロックを対象として、それらの後尾にある共通命令列集合を抽出する共通命令列集合抽出手段と、初回は同一の基本ブロックに無条件に後続する複数の基本ブロックを対象として共通命令列集合抽出手段に共通命令列集合を抽出させ、2回目以降は共通の命令列を有する複数の基本ブロックから直前までに抽出された共通の命令列を除外したプログラム部分を対象として共通命令列集合抽出手段に共通命令列集合を抽出させる抽出制御手段とを備えることを特徴とする請求項1記載のプログラム変換装置。

【請求項10】 前記選択手段は、

上下に連続する複数の階層にわたる共通命令列集合について、一対の共通命令列集合が上下に接する場合に、双方において同一の基本ブロックを選択することを特徴とする請求項1または2記載のプログラム変換装置。

【請求項11】 前記選択手段は、

上下に連続する複数の階層にわたる共通命令列集合について、下層の1の共通命令列集合に対し上層の複数の共通命令列集合が接する場合に、要素数が最も多い上層の共通命令列集合と下層の共通命令列集合とにおいて同一の基本ブロックを選択することを特徴とする請求項1または2記載のプログラム変換装置。

【請求項12】 前記選択手段は、

最後尾が分岐命令でない基本ブロックに含まれる命令列が共通命令列集合に属している場合に、当該基本ブロックを選択することを特徴とする請求項1または2記載のプログラム変換装置。

【請求項13】 前記選択手段は、

抽出された各共通命令列集合において1以上の基本ブロックを、他の共通命令列集合における基本ブロックとの位置関係に応じて選択候補とする選択候補抽出部と、各共通命令列集合において選択候補とされた基本ブロック

を1づつ選ぶ組合せを全て列挙する組合せ列挙部と、列挙された各々の組合せに従って基本ブロックを選択して前記変換を行った場合に変更される分岐命令のコードサイズをあらかじめ算出する分岐命令サイズ算出部とを備え、算出されるコードサイズが最も小さい組合せに従って基本ブロックを選択することを特徴とする請求項1または2記載のプログラム変換装置。

【請求項14】 前記変換手段は、さらに、前記変換により削除される命令列および変更される分岐命令についてコードサイズの変化量を算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換手段に対して制御する制御手段を備えることを特徴とする請求項1または2記載のプログラム変換装置。

【請求項15】 前記変換手段は、さらに、前記変換により削除される命令列および変更される分岐命令についてコードサイズの変化量を算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換手段に対して制御する制御手段を備えることを特徴とする請求項1または2記載のプログラム変換装置。

【請求項16】 前記変換手段は、さらに、前記変換により削除される命令列および変更される分岐命令についてコードサイズの変化量を算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換手段に対して制御する制御手段を備えることを特徴とする請求項1または2記載のプログラム変換装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータプログラムの変換技術に関し、特にプログラムの最適化技術に関する。

【0002】

【従来の技術】近年、ソフトウェア開発においては、開発効率、および保守性向上を指向して高水準言語を用いた開発が主流となっている。しかしながら、このような高水準言語からコンパイラが生成する目的プログラムは、アセンブラー等を用いて人間が作成する目的プログラムに比べると、一般に実行時間がより長くコードサイズがより大きい。このため、実行時間がより短くコードサイズがより小さいプログラムを生成することが、コンパイラに求められている。特に、記憶容量が厳しく制限される実行環境において稼動するプログラム、例えば組込み用途などのプログラムについては、コードサイズ削減への強い要請がある。

【0003】実行時間を短縮しつつコードサイズを削減する目的でプログラムを変換すること、または、実行時間またはコードサイズの一方を改善する目的でプログラムを変換することを最適化という。実行時間またはコードサイズの一方を改善する最適化においては、実行時間を短縮することによってコードサイズが増大する場合や、コードサイズを削減することによって実行時間が増加する場合があり得る。

【0004】最適化を行うための様々な技術が公知となっており、その中で、コードサイズを削減するものの1つに共通命令列合流最適化（テールマージ最適化ともいう）がある。テールマージ最適化とは、同一の基本ブロックに無条件に制御を移す複数の基本ブロックの後尾にある同一の命令列のうち、1の命令列が共用されるようにプログラムを変換してコードサイズを削減する技法である。前記同一の命令列を要素とする集合を、以下、共通命令列集合と称する。

【0005】ここで、基本ブロックとは、一連の命令列であって、途中の命令から分岐せずかつ途中の命令へ分岐されることもない最長の命令列をいう。図19は、テールマージ最適化前後のプログラムの例を示しており、1100aは最適化前のプログラム、1100bは最適化後のプログラム、1101a、1101b、1102a、1102b、1103、1104は基本ブロック、1105は共通命令列集合、1106は共用される命令列、1107～1110は分岐命令である。

【0006】従来のコンパイラは、この例のテールマージ最適化において、共通命令列集合1105について基本ブロック1102aに含まれる共通命令列を残し、基本ブロック1101aに含まれる共通命令列を削除し、分岐命令1107の分岐先を変更して前記残した命令列1106が実行されるよう、プログラムを変換する。この変換により、基本ブロック1101aから削除した共

通命令列に相当するコードサイズを削減できる。

【0007】しかしながら、この最適化を行うことにより、プログラムの実行時間が増加する。最適化前のプログラム1100aでは基本ブロック1101aを通る処理において1つの分岐命令1107が実行されるが、最適化後のプログラム1100bでは分岐命令1109、1110の2つが実行される。すなわち、最適化を行うことによって基本ブロック1101bを通る処理において実行される分岐命令数が1増加し、このためにプログラムの実行時間が増加する。

【0008】最適化後、基本ブロック1102bを通る処理において実行される分岐命令数は最適化前と変わらず、プログラム全体として分岐命令数は1増加する。図20は、テールマージ最適化を2回繰り返し行った過程を示すプログラム例である。1200aは最適化前のプログラム、1200bは1回目のテールマージ最適化後のプログラム、1200cは2回目のテールマージ最適化後のプログラム、1201a、1201b、1202a、1202b、1202c、1203a、1203b、1203c、1204、1205、1206は基本ブロック、1207～1208は共通命令列集合、1209～1210は共用される命令列、1211～1218は分岐命令である。

【0009】このように、テールマージ最適化を繰り返し行って命令列を共用させることを、多段階テールマージ最適化と呼ぶ。図20の多段階テールマージ最適化は、1回目に共通命令列集合1207について基本ブロック1201aに含まれる共通命令列が共用されるようにプログラムを変換し、2回目に共通命令列集合1208について基本ブロック1203bに含まれる共通命令列が共用されるようにプログラムを変換した例を示している。

【0010】この最適化後のプログラム1200cについて、基本ブロック1201b、1202c、1203cを通る処理において実行される分岐命令数は、それぞれ、1、3、2であり、合計6である。図21は、図20とは異なる基本ブロックに含まれる命令列が共用されるように多段階テールマージ最適化を行った過程を示すプログラム例である。1300aは最適化前のプログラム、1300bは1回目のテールマージ最適化後のプログラム、1300cは2回目のテールマージ最適化後のプログラム、1301a、1301b、1302a、1302b、1302c、1303a、1303b、1303c、1304、1305、1306は基本ブロック、1307～1308は共通命令列集合、1309～1310は共用される命令列、1311～1318は分岐命令である。

【0011】図21の多段階テールマージ最適化は、1回目に共通命令列集合1307について基本ブロック1303aに含まれる共通命令列が共用されるようにプロ

グラムを変換し、2回目に共通命令列集合1308について基本ブロック1303bに含まれる共通命令列が共用されるようにプログラムを変換した例を示している。

【0012】この最適化後のプログラム1300cについて、基本ブロック1301b、1302c、1303cを通る処理において実行される分岐命令数は、それぞれ、2、2、1であり、合計5である。このように、多段階テールマージ最適化においては、各段階において何れの基本ブロックに含まれる共通命令列が共用されるかの組合せに応じて、最適化後のプログラム全体として実行される分岐命令数が異なる。図20および図21の例の比較では、各段階で同一の基本ブロックに含まれる共通命令列が共用されるほうが（図21）異なる基本ブロックに含まれる共通命令列が共用されるよりも（図20）分岐命令数がより少なく抑えられる。

【0013】

【発明が解決しようとする課題】しかしながら、従来の多段階テールマージ最適化においては、図20に示したように最適化後のプログラムにおいて実行される分岐命令数を最小に抑えるような組合せによって命令列が共用されない場合がある。すなわち、必ずしも実行時間が最短となるプログラムが得られないという問題がある。

【0014】上記の問題に鑑み、本発明は、複数の基本ブロックの末尾にある同一の命令列についてそのうちの1が共用されるようにプログラムを変換してコードサイズを削減するプログラム変換装置であって、実行される分岐命令数が最小となるプログラムを生成する装置の提供を目的とする。

【0015】

【課題を解決するための手段】上記問題を解決するため、本発明のプログラム変換装置は、複数の基本ブロックの後尾にある同一の命令列についてそのうちの1が共用されるようにプログラムを変換してコードサイズを削減するプログラム変換装置であって、特定の基本ブロックに無条件に後続する他の複数の基本ブロックを対象として、それらの異なる基本ブロック内に存在する同一の命令列の集合であって、それらの基本ブロックの後尾から先頭に向けて階層的に存在する共通命令列集合を抽出する抽出手段と、抽出された各共通命令列集合において何れか1の命令列を含む基本ブロックを、他の共通命令列集合において選択した基本ブロックとの位置関係に応じて選択する選択手段と、選択された基本ブロック中の命令列が他の基本ブロックにおいて共用されるようにプログラムを変換するコード変換手段とを備える。

【0016】前記抽出手段は、複数の基本ブロックを対象として、それらの基本ブロックの後尾にある共通命令列集合を抽出する共通命令列集合抽出手段と、初回は同一の基本ブロックに無条件に後続する複数の基本ブロックを対象として共通命令列集合抽出手段に共通命令列集合を抽出させ、2回目以降は共通の命令列を有する複数

の基本ブロックから直前までに抽出された共通の命令列を除外したプログラム部分を対象として共通命令列集合抽出手段に共通命令列集合を抽出させる抽出制御手段とを備えてもよい。

【0017】前記選択手段は、上下に連続する複数の階層にわたる共通命令列集合について、一対の共通命令列集合が上下に接する場合に、双方において同一の基本ブロックを選択してもよい。前記選択手段は、上下に連続する複数の階層にわたる共通命令列集合について、下層の1の共通命令列集合に対し上層の複数の共通命令列集合が接する場合に、要素数が最も多い上層の共通命令列集合と下層の共通命令列集合とにおいて同一の基本ブロックを選択してもよい。

【0018】前記選択手段は、最後尾が分岐命令でない基本ブロックに含まれる命令列が共通命令列集合に属している場合に、当該基本ブロックを選択してもよい。前記選択手段は、抽出された各共通命令列集合において1以上の基本ブロックを、他の共通命令列集合における基本ブロックとの位置関係に応じて選択候補とする選択候補抽出部と、各共通命令列集合において選択候補とされた基本ブロックを1つ選ぶ組合せを全て列挙する組合せ列挙部と、列挙された各々の組合せに従って基本ブロックを選択して前記変換を行った場合に変更される分岐命令のコードサイズをあらかじめ算出する分岐命令サイズ算出部とを備え、算出されるコードサイズが最も小さい組合せに従って基本ブロックを選択してもよい。

【0019】前記プログラム変換装置は、さらに、前記変換により削除される命令列および変更される分岐命令についてコードサイズの変化量を算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換手段に対して制御する制御手段を備えてもよい。本発明のプログラム変換方法は、複数の基本ブロックの後尾にある同一の命令列についてそのうちの1が共用されるようにプログラムを変換してコードサイズを削減するプログラム変換装置において用いられるプログラム変換方法であって、特定の基本ブロックに無条件に後続する他の複数の基本ブロックを対象として、それらの異なる基本ブロック内に存在する同一の命令列の集合であって、それらの基本ブロックの後尾から先頭に向けて階層的に存在する共通命令列集合を抽出する第1のステップと、第1のステップで抽出された各共通命令列集合において何れか1の命令列を含む基本ブロックを、他の共通命令列集合において選択した基本ブロックとの位置関係に応じて選択する第2のステップと、第2のステップで選択された基本ブロック中の命令列が他の基本ブロックにおいて共用されるようにプログラムを変換する第3のステップとを含む。

【0020】本発明のプログラム記録媒体は、複数の基本ブロックの後尾にある同一の命令列についてそのうち

の1が共用されるようにプログラムを変換してコードサイズを削減するプログラム変換装置において用いられるプログラムを記憶しているコンピュータ読み取り可能な記録媒体であって、特定の基本ブロックに無条件に後続する他の複数の基本ブロックを対象として、それらの異なる基本ブロック内に存在する同一の命令列の集合であって、それらの基本ブロックの後尾から先頭に向けて階層的に存在する共通命令列集合を抽出する第1のステップと、第1のステップで抽出された各共通命令列集合において何れか1の命令列を含む基本ブロックを、他の共通命令列集合において選択した基本ブロックとの位置関係に応じて選択する第2のステップと、第2のステップで選択された基本ブロック中の命令列が他の基本ブロックにおいて共用されるようにプログラムを変換する第3のステップとを有するプログラムを記憶している。

【0021】

【発明の実施の形態】<第1実施形態>第1実施形態における、本発明のプログラム変換装置について説明する。

<全体構成>本発明のプログラム変換装置100は、ソースプログラムを目的プログラムに変換する装置であり、図1に示すように、字句解析部102、構文解析部103、中間コード生成部104、中間コード最適化部105、および、目的プログラム生成部107から構成される。

【0022】プログラム変換装置100は、具体的には、プロセッサ、プログラムを記憶しているROM(Read Only Memory)、作業用のRAM(Random Access Memory)等のハードウェアにより実現される。前記各部の機能は、ROMに記憶されているプログラムをプロセッサが実行することにより実現される。前記各部の間におけるデータの受け渡しは、RAM等のハードウェアを介して行われる。

【0023】字句解析部102は、入力されたソースプログラム101を解析し、ソースプログラム101を構成する字句を認識する。構文解析部103は、字句解析部102が認識した字句に基づいてソースプログラム101の構文を解析する。中間コード生成部104は、構文解析部103が解析した構文の意味を中間言語を用いて表記した中間コードを生成する。中間コード最適化部105は、中間コード生成部104が生成した中間コードを以下に示すようにして最適化する。目的プログラム生成部107は、中間コード最適化部105が最適化した中間コードに従って目的プログラム106を生成する。

【0024】中間コード最適化部105以外の構成は、一般的なコンパイラにおいて広く実施されている公知技術であるため、ここでは、詳細な説明を省略する。

<中間コード最適化部>図2は、中間コード最適化部1

05を示すブロック図である。同図に示すように、中間コード最適化部105は、基本ブロック解析部111、および、テールマージ最適化部112からなる。

【0025】基本ブロック解析部111は、中間コード生成部104が生成した中間コードを解析して、基本ブロック、基本ブロック間の制御フロー、および、同一の基本ブロックに無条件に制御を移すような複数の基本ブロックを要素とする集合(以後、無条件先行ブロック集合と称する)と後続の基本ブロックとの組を抽出する。

【0026】図3(a)は基本ブロック解析部111によって抽出された基本ブロックおよび基本ブロック間の制御フローの模式図、図3(b)は無条件先行ブロック集合の一覧を示す一例であり、201～207は基本ブロック、211～219は基本ブロック間の制御フロー、220は無条件先行ブロック集合と後続の基本ブロックとの組の一覧を表わしている。

【0027】基本ブロック解析部111が行うこれらの処理は、一般的なコンパイラにおいて広く実施されている公知技術であるため、ここでは、詳細な説明を省略する。テールマージ最適化部112は、共通命令列集合抽出部114、共用部分選択部115、および、コード変換部116からなる。

<共通命令列集合抽出部>共通命令列集合抽出部114は、基本ブロック解析部111によって抽出された基本ブロック、制御フロー、および、無条件先行ブロック集合に基づいて、再帰的な抽出処理を含む抽出準備処理を行うことにより、無条件先行ブロック集合の後尾から先頭に向けて階層的に存在する共通命令列集合を抽出する。

<抽出準備処理>図4は、抽出準備処理を示すフローチャートである。図6は本処理の過程を示すために用いるプログラム(中間コード)の一例であり、300は無条件先行ブロック集合、301～305は無条件先行ブロック集合300に属する基本ブロック、306は後続の基本ブロック、307～309は共通命令列集合、311～315は命令である。

【0028】以下、図4のフローチャートおよび図6のプログラム例を参照しながら、抽出準備処理について説明する。共通命令列集合抽出部114は、ステップS10～ステップS13の繰返し処理により、基本ブロック解析部111によって抽出された無条件先行ブロック集合ごとに、無条件先行ブロック集合に属する各基本ブロックの分岐命令を除いた最後尾の命令を求め(ステップS11)、当該求めた命令を指定して抽出処理を呼び出す(ステップS12)。

【0029】図6の例では、無条件先行ブロック集合300に属する基本ブロック301～305の分岐命令を除いた最後尾の命令である命令315、325、335、345、355を指定され、抽出処理が呼び出される。

<抽出処理>図5は、ステップS12の抽出処理の詳細を示すフローチャートである。以下、図5のフローチャートおよび図6のプログラム例を参照しながら、抽出処理について説明する。

【0030】初回の抽出処理において、共通命令列集合抽出部114は、ステップS12の処理において指定された命令を同じ種類ごとに分類する（ステップS20）。本例では、全てmov R0, (_a) 命令である5個の命令が指定されており、1種類に分類される。共通命令列集合抽出部114は、ステップS21～ステップS25の繰返し処理により、前記分類され、同じ命令が2以上ある命令の種類ごとに、当該種類の命令から基本ブロックの先頭に向かって命令の異同を順次比較し、全ての基本ブロックにおいて同じ命令が連続する部分を求め（ステップS22）、当該求めた部分を要素とする共通命令列集合を作成する（ステップS23）。

【0031】図6の例では、mov R0, (_a) 命令について、命令315、325、335、345、355から基本ブロックの先頭に向かって命令の異同が順次比較され、基本ブロック301～305において同じ命令が連続する部分である、命令314～315、命令324～325、命令334～335、命令344～345、命令354～355を要素とする共通命令列集合307が作成される。

【0032】共通命令列集合抽出部114は、作成された共通命令列集合の各要素の直前に位置する命令を指定して図5のフローチャートに示す抽出処理を再帰的に呼び出し、基本ブロックの先頭に向かって階層的に存在する共通命令列集合を繰返し抽出する（ステップS24）。図6の例では、作成された共通命令列集合307の各要素の直前に位置する命令313、323、333、343、353を指定され2回目の抽出処理が呼び出される。

【0033】2回目の抽出処理において、共通命令列集合抽出部114は、指定された命令313、323、333、343、353をmov (_d), R1命令である命令313、323、および、mov (_c), R1命令である命令333、343、353の2種類に分類する（ステップS20）。mov (_d), R1命令について、命令312～313、命令322～323を要素とする共通命令列集合308が作成され、当該各要素の直前に位置する命令311、321を指定されて3回目の抽出処理が呼び出される。

【0034】3回目の抽出処理において、共通命令列集合抽出部114は、前記指定された命令311、312をmov (_a), R1命令である命令311と、mov (_b), R1命令である命令312とに分類する（ステップS20）。何れの種類の命令も1個であり、ステップS21～ステップS25の繰返し処理の条件を満たさないため、命令311、321から基本ブロック

の先頭に向かう部分について共通命令列集合の抽出処理は行われず、2回目の抽出処理の続きを戻る。

【0035】2回目の抽出処理の続きにおいて、mov (_c), R1命令について、命令332～333、命令342～343、命令352～353を要素とする共通命令列集合309が作成され、作成された共通命令列集合309の直前に位置する命令331、341、351を指定されて4回目の抽出処理が呼び出される。4回目の抽出処理においても、3回目と同様に、命令331、341、351から基本ブロックの先頭に向かう部分について共通命令列集合の抽出処理は行われない。

【0036】この結果、無条件先行ブロック集合300において、共通命令列集合307～309が抽出される。

<共用部分選択部>共用部分選択部115は、再帰的な選択処理を含む選択準備処理を行うことにより、共通命令列集合抽出部114によって抽出された各共通命令列集合において共用されるべき1の命令列を含む基本ブロックを、他の共通命令列集合において選択された基本ブロックとの位置関係に応じて選択する。

<選択準備処理>図7は、選択準備処理を示すフローチャートである。以下、図7のフローチャートおよび図6のプログラム例を参照しながら、選択準備処理について説明する。

【0037】共用部分選択部115は、ステップS30～ステップS35の繰返し処理により、基本ブロック解析部111によって抽出された無条件先行ブロック集合ごとに、無条件先行ブロック集合の分岐命令を除いた最後尾に位置する共通命令列集合を求める（ステップS31）、さらにステップS32～ステップS34の繰返し処理により、当該求めた共通命令列集合を順次指定して選択処理を行う（ステップS33）。

【0038】図6の例では、無条件先行ブロック集合300の最後尾の共通命令列集合307を指定され、選択処理が呼び出される。

<選択処理>図8は、ステップS33の選択処理の詳細を示すフローチャートである。以下、図8のフローチャートおよび図6のプログラム例を参照しながら、選択処理について説明する。

【0039】選択処理の前半において、共用部分選択部115は、指定された共通命令列集合の直前に別の共通命令列集合が存在する場合（ステップS40）、ステップS41～ステップS43の繰返し処理により、当該直前に存在する各共通命令列集合を順次指定して、図8のフローチャートに示す選択処理を再帰的に呼び出す（ステップS42）。この呼び出しが再帰的に繰返されることにより、指定された共通命令列集合において基本ブロックを選択するよりも前に、当該共通命令列集合から先頭に向かって階層的に存在する全ての共通命令列集合において共用されるべき命令列を含む基本ブロックが予め選

択される。

【0040】選択処理の後半において、共用部分選択部115は、指定された共通命令列集合において、以下の条件判断に従って共用されるべき命令列を含む基本ブロックを選択する。すなわち、指定された共通命令列集合に最後尾が分岐命令でない基本ブロックが属している場合は（ステップS44）当該基本ブロックを選択する（ステップS45）。指定された共通命令列集合の直前に複数の共通命令列集合が存在する場合は（ステップS46）当該複数の共通命令列集合のうち要素数が最大のものにおいて選択した基本ブロックを選択する（ステップS47）。指定された共通命令列集合の直前に1の共通命令列集合が存在する場合は（ステップS48）当該共通命令列集合において選択した基本ブロックを選択する（ステップS49）。何れの場合にも該当しない場合は指定された共通命令列集合に属する任意の1の基本ブロックを選択する（ステップS50）。

【0041】図6の例では、初回の選択処理は共通命令列集合307が指定されて呼び出され、当該処理の前半において、共通命令列集合307の直前に位置する共通命令列集合308、309を各々指定して、2回目、3回目の選択処理が呼び出される。2回目の選択処理の前半において、指定された共通命令列集合308の直前に位置する共通命令列集合が存在しないため、選択処理の再帰的な呼び出しは行われない。同処理の後半において、共通命令列集合308において共用されるべき命令列を含む基本ブロックが選択される。この場合はステップS44、S46、S48の何れの条件にも該当せず、ステップS50に従い任意の1の基本ブロックが選択される。ここでは、命令312～313を含む基本ブロック301が選択されるものとする。

【0042】3回目の選択処理においても、指定された共通命令列集合309について、2回目と同様の処理および条件判断が行われる。ここでは、命令332～333を含む基本ブロック303が選択されるものとする。2回目、3回目の選択処理が行われた後、初回の選択処理の後半が実行される。共通命令列集合307の直前に、共通命令列集合308、309が存在しステップS46の条件を満たすため、ステップS47により共通命令列集合309について選択した基本ブロック303を選択する。

【0043】この結果、共通命令列集合307、308、309において共用すべき命令列を含む基本ブロックとして、それぞれ、基本ブロック303、303、301が選択される。

<選択条件の根拠>以下、前記ステップS44～S50により選択された基本ブロックに含まれる命令列を共用させると、最適化後のプログラムにおいて分岐命令の実行回数を最小にできることを、図13～図16の概念図を用いて示す。

【0044】テールマージ最適化においてコンパイラは、各共通命令列集合において選択されなかった基本ブロックの命令列を選択された基本ブロックの命令列の先頭への分岐命令に置き換えることにより当該選択された基本ブロックの命令列を共用させる。このため、選択されなかった基本ブロックの命令列を元実行していた処理において当該置き換えられた分岐命令が実行される。係る分岐命令が実行される数は次のように求められる。

10 【0045】図13（a）は、共通命令列集合aの直前に他の共通命令列集合が存在せず、共通命令列集合aがm個の要素からなる場合を表わす概念図である。図13（b）、（c）にそれぞれ示すように、基本ブロック1の命令列を共用させると（m-1）個、基本ブロックmの命令列を共用させると（m-1）個の処理において、置き換えられた分岐命令 j m p L1 が実行される。分岐命令が実行される数はどの基本ブロックに含まれる命令列を選択しても（m-1）であり差異がない。ゆえに、ステップS50において、任意の1の基本ブロックを選択する。

20 【0046】図14（a）は、共通命令列集合aの直前に1の共通命令列集合bが存在して、共通命令列集合aがm個、共通命令列集合bがi個（1 < i < m）の要素からなる場合を表わす概念図である。図14（b）、（c）にそれぞれ示すように、共通命令列集合aにおいて共用させた命令列aと同一の基本ブロックに含まれる命令列を共用させると（m-i）個、それ以外の命令列を共用させると（m-1）個の処理において、置き換えられた分岐命令 j m p L1 が実行される。ここで（m-i） < (m-1) であり、直前の共通命令列集合において共用させた命令列と同一の基本ブロックに含まれる命令列を共用させると、実行される分岐命令の数をより少なくできる。ゆえに、ステップS48、S49において、直前の共通命令列集合において選択した基本ブロックを選択する。

30 【0047】図15（a）は、共通命令列集合aの直前に複数の共通命令列集合b、c、…が存在して、共通命令列集合aがm個、共通命令列集合b、c、…がi、j、…個（1 < … < j < i < m）の要素からなる場合を表わす概念図である。図15（b）、（c）、（d）に

40 それぞれ示すように、共通命令列集合bにおいて共用させた命令列の直後の命令列を共用させると（m-i）個、共通命令列集合cにおいて共用させた命令列の直後の命令列を共用させると（m-j）個、それ以外の命令列を共用させると（m-1）個の処理において、置き換えられた分岐命令 j m p L1 が実行される。ここで（m-i） < (m-j) < (m-1) であり、直前の共通命令列集合のうち要素数が最も多いものにおいて共用させた命令列と同一の基本ブロックに含まれる命令列を共用させると、分岐命令の実行回数をより少なくできる。ゆえに、ステップS46、S47において、直前の

13

共通命令列集合のうち要素数が最大のものについて選択した基本ブロックを選択する。

【0048】前記命令列を共用させるための分岐命令のほかに、後続の基本ブロックへ制御を移行させるための分岐命令が実行される場合がある。係る分岐命令は、後続の基本ブロックとメモリ領域が連続している高々1個の基本ブロックを除いて全ての基本ブロックの最後尾に存在する。図16(a)は、無条件先行ブロック集合にm個の基本ブロックが属しており、基本ブロック1の最後尾の命令が分岐命令でない場合を表わす概念図である。図16(b)、(c)にそれぞれ示すように、基本ブロック1に含まれる命令列を共用させると係る分岐命令jmpl0は全く実行されず、その他の基本ブロックに含まれる命令列を共用させるとm個の処理において係る分岐命令jmpl0が実行される。

【0049】この個数mは、前記命令列を共用させるために実行される分岐命令の個数の何れよりも大きいため、他の条件に優先してこの分岐命令を含まない命令列を選択すると分岐命令の実行回数をより少なくできる。ゆえに、ステップS44、S45において、最後尾が分岐命令でない基本ブロックが共通命令列集合に属している場合は、他の条件に優先して当該基本ブロックを選択する。

【0050】以上より、ステップS44～S50で選択された基本ブロックに含まれる命令列を共用させることで、最適化後のプログラムにおいて分岐命令の実行回数を最小にできる。

<コード変換部>コード変換部116は、再帰的なコード変換処理を含むコード変換準備処理を実行して、共用部分選択部115が選択した基本ブロックに含まれる命令列が各共通命令列集合において共用されるよう、プログラムを変換する。

<コード変換準備処理>図9は、コード変換準備処理を示すフローチャート、図11および図12はコード変換準備処理およびコード変換処理の過程を示すために用いるプログラムの一例である。図11は図6のプログラムに対して共通命令列集合308、309について共用部分選択部115が選択した基本ブロックに含まれる命令列を共用させた結果のプログラムを示しており、301a、301b、302a、303a、303b、304a、305a、306は基本ブロック、307、308、309は共通命令列集合、311～325、326a、331～345、346a、351～355、356aは命令およびラベルである。図12は図11のプログラムに対して共通命令列集合307について共用部分選択部115が選択した基本ブロックに含まれる命令列を共用させた結果のプログラムを示しており、301a、301c、302a、303a、303b、303c、304a、305a、306は基本ブロック、307、308、309は共通命令列集合、311～31

14

3、316a、317、321、326a、331～341、346a、351、356aは命令およびラベルである。

【0051】以下、図9のフローチャートおよび図11のプログラム例を参照しながら、コード変換準備処理について説明する。コード変換部116は、ステップS60からステップS65の繰返し処理により、基本ブロック解析部111によって抽出された無条件先行ブロック集合ごとに、無条件先行ブロック集合の分岐命令を除了した最後尾の共通命令列集合を求め(ステップS61)、さらにステップS62～ステップS64の繰返し処理により、当該求めた共通命令列集合を順次指定してコード変換処理を行う(ステップS63)。

【0052】図11の例では、無条件先行ブロック集合300の最後尾に位置する共通命令列集合307を指定され、コード変換処理が呼び出される。

<コード変換処理>図10は、ステップS63のコード変換処理の詳細を示すフローチャートである。以下、図10のフローチャート並びに図11および図12のプログラム例を参照しながら、コード変換処理について説明する。

【0053】コード変換処理の前半において、コード変換部116は、指定された共通命令列集合の直前に別の共通命令列集合が存在する場合(ステップS70)、ステップS71～ステップS73の繰返し処理により、当該直前に存在する各共通命令列集合を順次指定して、図10のフローチャートに示すコード変換処理を再帰的に呼び出す(ステップS72)。この呼び出しが再帰的に繰返されることによって、指定された共通命令列集合においてコード変換を行うよりも前に、当該共通命令列集合から先頭に向けて階層的に存在する全ての共通命令列集合において共用部分選択部115によって選択された基本ブロックに含まれる命令列が共用されるようにプログラムが変換される。

【0054】コード変換処理の後半において、コード変換部116は、共用部分選択部115によって選択された基本ブロックに含まれる命令列の直前にラベルがない場合は(ステップS74)、当該場所に新たにラベルを作成し(ステップS75)、その後ステップS76からステップS80の繰返し処理により、選択されなかった各基本ブロックについて順次以下の処理を行う。すなわち、当該基本ブロックの最後尾の分岐命令について分岐先が更新されていなければ(ステップS77)前記選択された基本ブロックの命令列の直前に変更し(ステップS78)、当該選択されなかった基本ブロックの命令列を削除する(ステップS79)。

【0055】図11および図12の例では、初回のコード変換処理は共通命令列集合307を指定して呼び出され、当該処理の前半において、共通命令列集合307の直前に位置する共通命令列集合308、309を各々指

定して、2回目、3回目のコード変換処理が呼び出される。2回目のコード変換処理の前半において、指定された共通命令列集合308の直前に位置する他の共通命令列集合が存在しないため、コード変換処理の再帰的な呼び出しは行われない。同処理の後半において、共用部分選択部115によって選択された基本ブロック301aに含まれる命令312～313からなる命令列の直前に新たにラベル317が作成され、分岐命令326aの分岐先がL2に変更され、選択されなかった基本ブロックの命令列が削除される。

【0056】3回目のコード変換処理の前半においても、2回目の処理と同様、コード変換処理の再帰的な呼び出しは行われず、同処理の後半において、共用部分選択部115によって選択された基本ブロック301bに含まれる命令332～333からなる命令列の直前に新たにラベル337が作成され、分岐命令346a、356aの分岐先がL3に変更され、選択されなかった基本ブロックの命令列が削除される。

【0057】この結果、図11に示したプログラムが得られる。2回目、3回目のコード変換処理が行われた後、初回のコード変換処理の後半が実行される。共用部分選択部115によって選択された基本ブロック303cに含まれる命令334～335からなる命令列の直前に新たにラベル338が作成され、まだ変更されていない分岐命令316aの分岐先がL4に変更され、選択されなかった基本ブロックの命令列が削除される。

【0058】この結果、図12に示したプログラムが得られる。この構成によれば、多段階テールマージ最適化後のプログラムにおいて実行される分岐命令の数を最小にすることができる。なお、前記共通命令列集合抽出部114に共通命令列集合抽出手段および共通命令列集合抽出制御手段を設け、抽出処理のステップS22～S23を共通命令列集合抽出手段により実行し、抽出処理の他のステップと抽出準備処理の各ステップとを共通命令列集合抽出制御手段により実行してもよい。

【0059】なお、前記共用部分選択部115の選択処理において、任意に1の命令列を選択すべき共通命令列集合、および、要素数が最大の共通命令列集合が直前に複数存在する共通命令列集合においては、共用すべき命令列の候補が複数存在する。係る共通命令列集合を含む無条件先行ブロック集合では分岐命令の実行回数を最小にする命令列の組合せが複数存在する。

【0060】係る場合には、前記共用部分選択部115は、各共通命令列集合において共用すべき命令列の候補を1つずつ選ぶ組合せをあらかじめ列挙し、各々の組合せに従って命令列を共用させた場合のプログラムについて分岐命令のサイズを算出し、サイズが最も小さくなる組合せを選択してもよい。この構成によれば、多段階テールマージ最適化後のプログラムにおいて実行される分岐命令の数を最小にした上で、分岐命令のサイズを最小

にすることができる。

【0061】なお、目的プログラムによっては分岐命令のコードサイズが分岐距離に応じて増減するという特性を有するものがある。係る特性を有する目的プログラムを生成する場合において、本プログラム変換装置は、コード変換部116が前記変換を行った場合に削除される命令列および変更される分岐命令についてコードサイズの変化量をあらかじめ算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換部116に対して制御する制御手段を備えてもよい。

【0062】この構成によれば、例えば、図12の基本ブロック301cの最後尾の分岐命令について分岐先が変更されることによりコードサイズが増加する場合であって、当該増加するコードサイズが削除される命令列のコードサイズと等しいかより大きい場合、基本ブロック301cにおける命令列の削除と分岐命令の変更は行われない。これにより、前記特性を有する目的プログラムにおいて、変換によりコードサイズが増加する可能性を排除できる。

【0063】なお、コンパイラによっては、生成された目的プログラムに対して最適化を行うものがある。係るコンパイラにおいては、目的プログラムに対してテールマージ最適化を行う構成としてもよい。図17は、係るプログラム変換装置120の構成を示す図であり、目的プログラム最適化部108は、基本ブロック解析部121、テールマージ最適化部122を含み、目的プログラム生成部107が生成した目的プログラムを入力され、前記中間コード最適化部105と同様の処理を行い、目的プログラム106を出力する。

【0064】この構成によれば、目的プログラムに対して最適化を行うコンパイラにおいても、本発明を適用することができる。

<第2実施形態>第2実施形態におけるプログラム変換装置は、第1実施形態に対して、共用部分選択部115の動作が異なる。第2実施形態において、共用部分選択部115は、選択した命令列を共用させたプログラムにおける分岐命令のサイズを算出する。当該分岐命令のサイズは共用させる命令列が含まれる基本ブロックの位置関係に応じて異なり、共用部分選択部115はサイズが最も小さくなる命令列を共用すべき命令列として選択する。

<選択処理>図18は、第2実施形態における選択処理を示すフローチャートである。以下、図18のフローチャートを参照しながら、第2実施形態における選択処理について説明する。

【0065】共用部分選択部115は、ステップS90～ステップS94の繰返し処理により、基本ブロック解析部111によって抽出された無条件先行ブロック集合ごとに、共用させた場合に分岐命令のサイズが最小とな

る命令列の組合せを求める。具体的には、各共通命令列集合から基本ブロックを1づつ選ぶ組合せを列挙し（ステップS91）、ステップS92～S94の繰返し処理により、列挙された組合せごとに、当該組合せに従って選択された基本ブロックに含まれる命令列を共用させた場合のプログラムにおける分岐命令のサイズを算出し、分岐命令のサイズが最も小さくなる組合せを共用すべき命令列が含まれる基本ブロックとして選択する（ステップS95）。

【0066】この構成によれば、多段階テールマージ最適化後のプログラムにおいて分岐命令のサイズを最小にすることができます。なお、生成された目的プログラムに対して最適化を行うコンパイラにおいては、目的プログラムに対してテールマージ最適化を行う構成としてもよい。この構成によれば、目的プログラムに対して最適化を行うコンパイラにおいても、本発明を適用することができます。

【0067】

【発明の効果】本発明のプログラム変換装置は、複数の基本ブロックの後尾にある同一の命令列についてそのうちの1が共用されるようにプログラムを変換してコードサイズを削減するプログラム変換装置であって、特定の基本ブロックに無条件に後続する他の複数の基本ブロックを対象として、それらの異なる基本ブロック内に存在する同一の命令列の集合であって、それらの基本ブロックの後尾から先頭に向けて階層的に存在する共通命令列集合を抽出する抽出手段と、抽出された各共通命令列集合において何れか1の命令列を含む基本ブロックを、他の共通命令列集合において選択した基本ブロックとの位置関係に応じて選択する選択手段と、選択された基本ブロック中の命令列が他の基本ブロックにおいて共用されるようにプログラムを変換するコード変換手段とを備える。

【0068】この構成によれば、あらかじめ、階層的に存在する全ての共通命令列集合を抽出し、共用させる命令列を含む基本ブロックの位置関係を複数の階層にわたって検討し、最適化を行ったプログラムにおいて実行される分岐命令数の多寡を知った上で、共用すべき命令列が含まれる基本ブロックを選択できるため、従来の多段階テールマージ最適化に比べて、当該分岐命令数をより低減することができる。

【0069】前記抽出手段は、複数の基本ブロックを対象として、それらの基本ブロックの後尾にある共通命令列集合を抽出する共通命令列集合抽出手段と、初回は同一の基本ブロックに無条件に後続する複数の基本ブロックを対象として共通命令列集合抽出手段に共通命令列集合を抽出させ、2回目以降は共通の命令列を有する複数の基本ブロックから直前までに抽出された共通の命令列を除外したプログラム部分を対象として共通命令列集合抽出手段に共通命令列集合を抽出させる抽出制御手段と

を備えてもよい。

【0070】この構成によれば、前記階層的に存在する共通命令列集合を効率よく抽出することができる。前記選択手段は、上下に連続する複数の階層にわたる共通命令列集合について、一対の共通命令列集合が上下に接する場合に、双方において同一の基本ブロックを選択してもよい。

【0071】前記選択手段は、上下に連続する複数の階層にわたる共通命令列集合について、下層の1の共通命令列集合に対し上層の複数の共通命令列集合が接する場合に、要素数が最も多い上層の共通命令列集合と下層の共通命令列集合とにおいて同一の基本ブロックを選択してもよい。前記選択手段は、最後尾が分岐命令でない基本ブロックに含まれる命令列が共通命令列集合に属している場合に、当該基本ブロックを選択してもよい。

【0072】これらの構成によれば、最適化を行ったプログラムにおいて実行される分岐命令数を最小にすることができる。前記選択手段は、抽出された各共通命令列集合において1以上の基本ブロックを、他の共通命令列集合における基本ブロックとの位置関係に応じて選択候補とする選択候補抽出部と、各共通命令列集合において選択候補とされた基本ブロックを1づつ選ぶ組合せを全て列挙する組合せ列挙部と、列挙された各々の組合せに従って基本ブロックを選択して前記変換を行った場合に変更される分岐命令のコードサイズをあらかじめ算出する分岐命令サイズ算出部とを備え、算出されるコードサイズが最も小さい組合せに従って基本ブロックを選択してもよい。

【0073】この構成によれば、最適化を行ったプログラムにおいて実行される分岐命令数を最小にした上で、分岐命令のサイズを最小にすることができます。前記プログラム変換装置は、さらに、前記変換により削除される命令列および変更される分岐命令についてコードサイズの変化量を算出し、変換によってコードサイズが減少しない基本ブロックにおいて命令列の削除および分岐命令の変更をしないようコード変換手段に対して制御する制御手段を備えてもよい。

【0074】この構成によれば、生成する目的プログラムの分岐命令のコードサイズが分岐距離に応じて増減するという特性を有する場合に、最適化を行うことによりコードサイズが増加する可能性を排除できる。

【図面の簡単な説明】

【図1】プログラム変換装置の機能ブロック図である。

【図2】中間コード最適化部の機能ブロック図である。

【図3】基本ブロックおよび基本ブロック間の制御フローの模式図、並びに、無条件先行ブロック集合の一覧を示す一例である。

【図4】抽出準備処理を示すフローチャートである。

【図5】抽出処理を示すフローチャートである。

【図6】抽出処理の過程を示すために用いるプログラム

の一例である。

【図7】選択準備処理を示すフローチャートである。

【図8】選択処理を示すフローチャートである。

【図9】コード変換準備処理を示すフローチャートである。

【図10】コード変換処理を示すフローチャートである。

【図11】コード変換処理の過程を示すために用いるプログラムの一例である。

【図12】コード変換処理の過程を示すために用いるプログラムの一例である。

【図13】選択条件の根拠を示すために用いる概念図である。

【図14】選択条件の根拠を示すために用いる概念図である。

【図15】選択条件の根拠を示すために用いる概念図である。

【図16】選択条件の根拠を示すために用いる概念図である。

【図17】プログラム変換装置の機能ブロック図である。

【図18】第2実施形態における選択処理を示すフローチャートである。

【図19】テールマージ最適化前後のプログラムの例である。

【図20】テールマージ最適化を2回繰返し行った過程を示すプログラム例である。

【図21】テールマージ最適化を2回繰返し行った過程を示すプログラム例である。

【符号の説明】

100 プログラム変換装置

101 ソースプログラム

102 字句解析部

103 構文解析部

104 中間コード生成部

105 中間コード最適化部

106 目的プログラム

107 目的プログラム生成部

108 目的プログラム最適化部

111 基本ブロック解析部

112 テールマージ最適化部

114 共通命令列集合抽出部

115 共用部分選択部

116 コード変換部

120 プログラム変換装置

121 基本ブロック解析部

122 テールマージ最適化部

300 無条件先行ブロック集合

301～305、301a～305a、301b、301c、303c 基本ブロック

307～309 共通命令列集合

311～356、316a、326a、346a、356a 命令

317、337、338 ラベル

1100a、1100b プログラム

1101a、1101b、1102a、1102b、1103a 基本ブロック

103、1104 基本ブロック

1105 共通命令列集合

1106 共用される命令列

1107～1110 分岐命令

1200a、1200b、1200c プログラム

1201a、1201b、1202a、1202b、1203a、1203b、1203c、1204、1205、1206 基本ブロック

1207、1208 共通命令列集合

1209、1210 共用される命令列

1211～1218 分岐命令

1300a、1300b、1300c プログラム

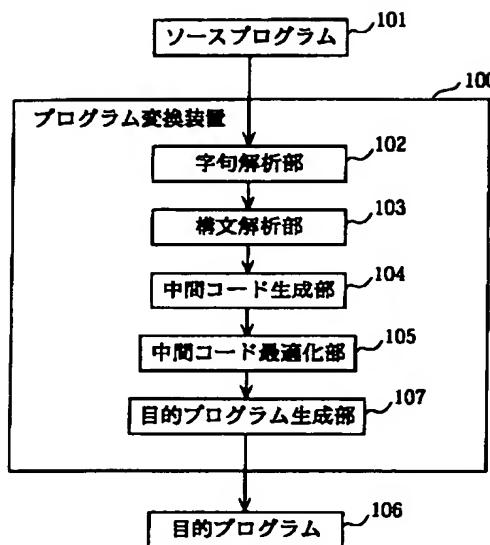
1301a、1301b、1302a、1302b、1303a、1303b、1303c、1304、1305、1306 基本ブロック

1307、1308 共通命令列集合

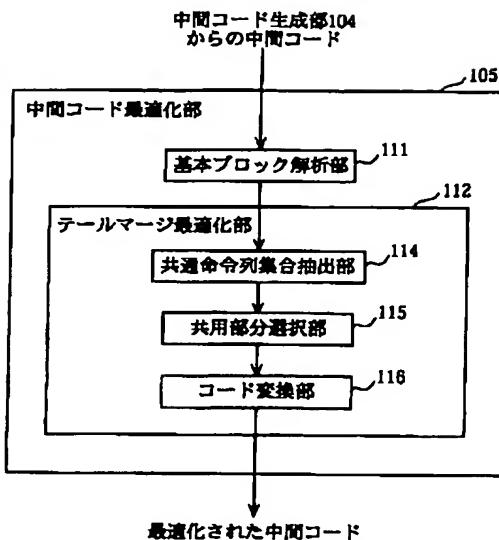
1309、1310 共用される命令列

1311～1318 分岐命令

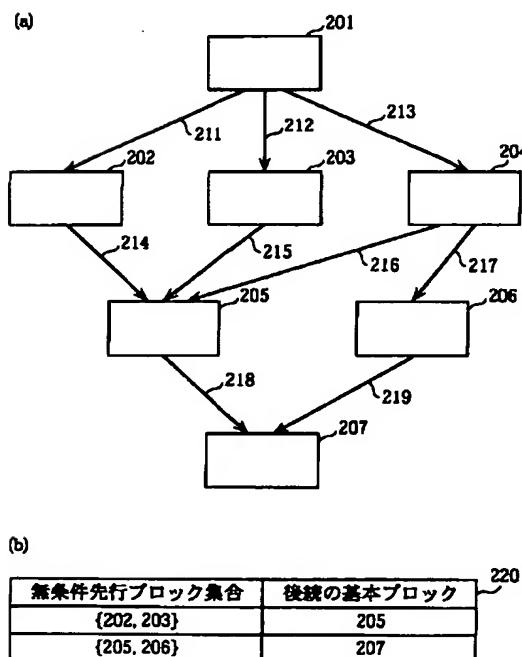
【図1】



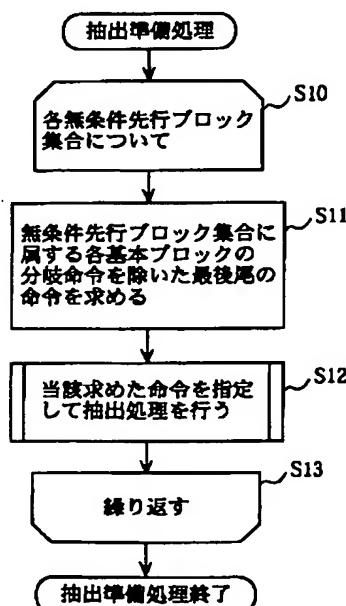
【図2】



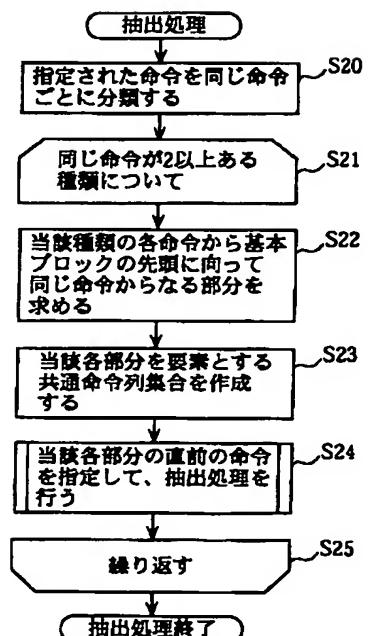
【図3】



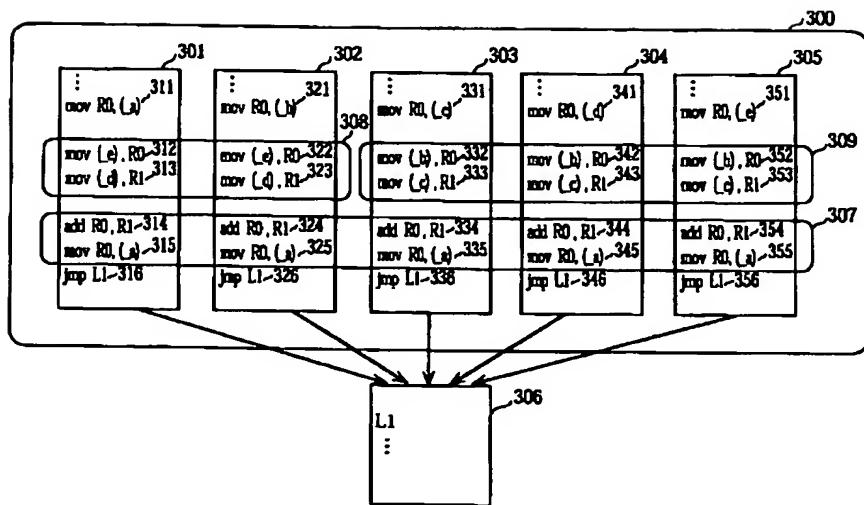
【図4】



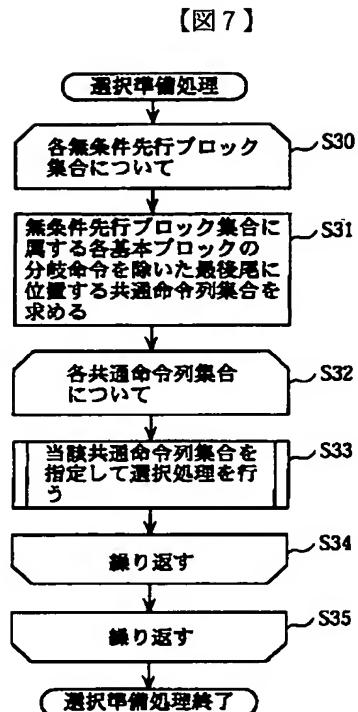
【図5】



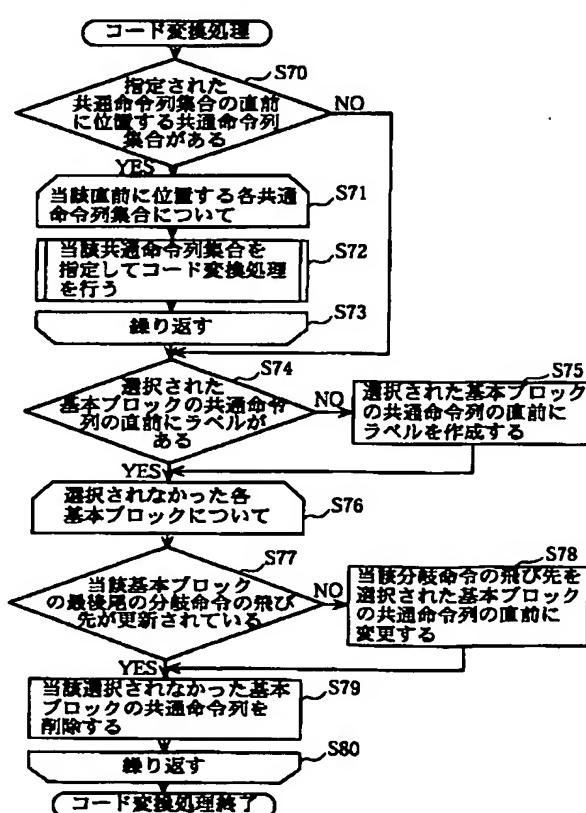
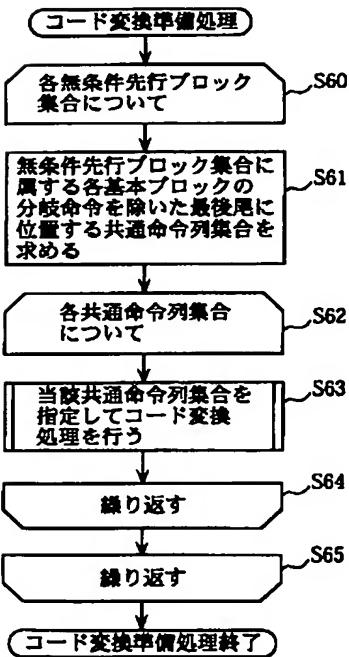
【図6】



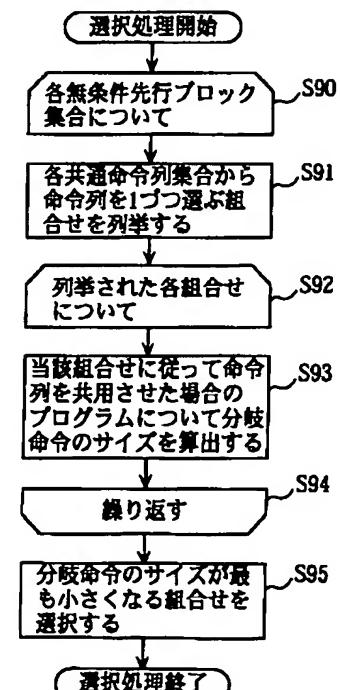
【図9】



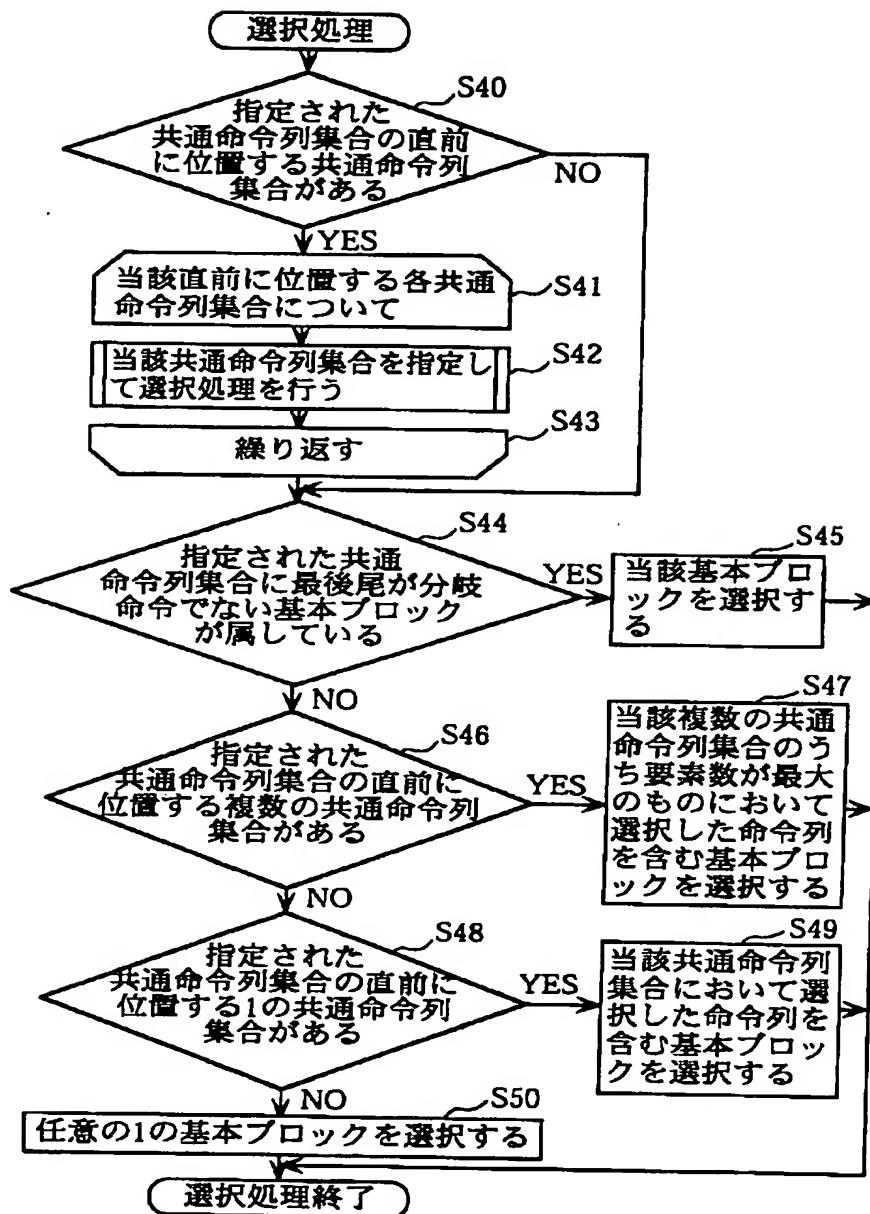
【図10】



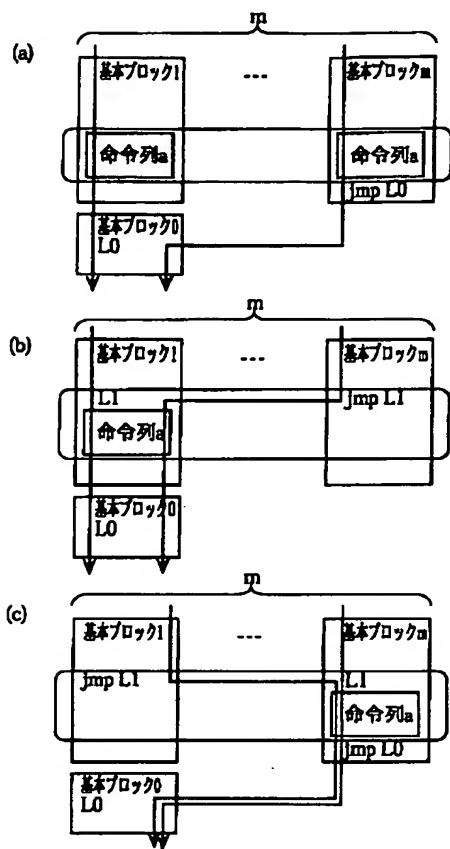
【図18】



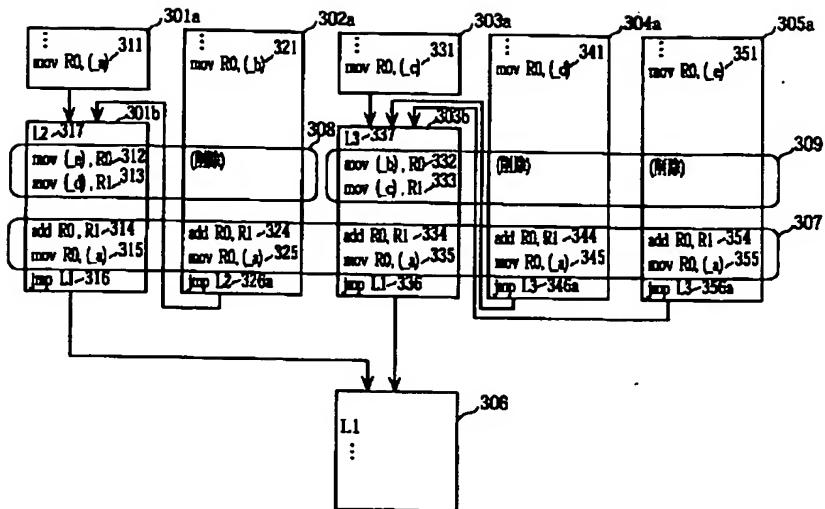
【図8】



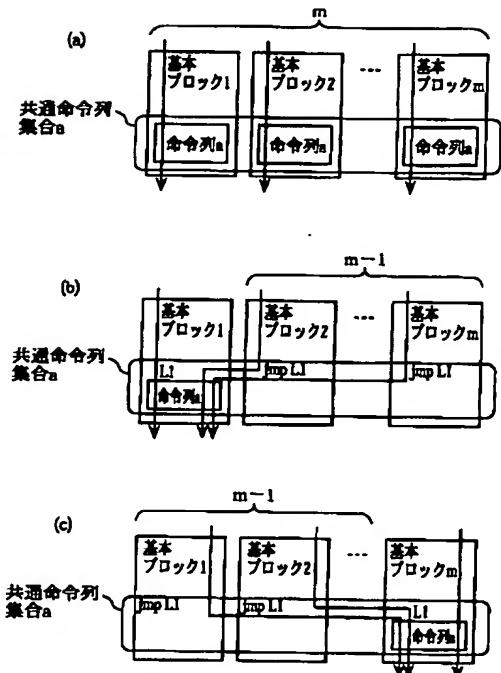
【図16】



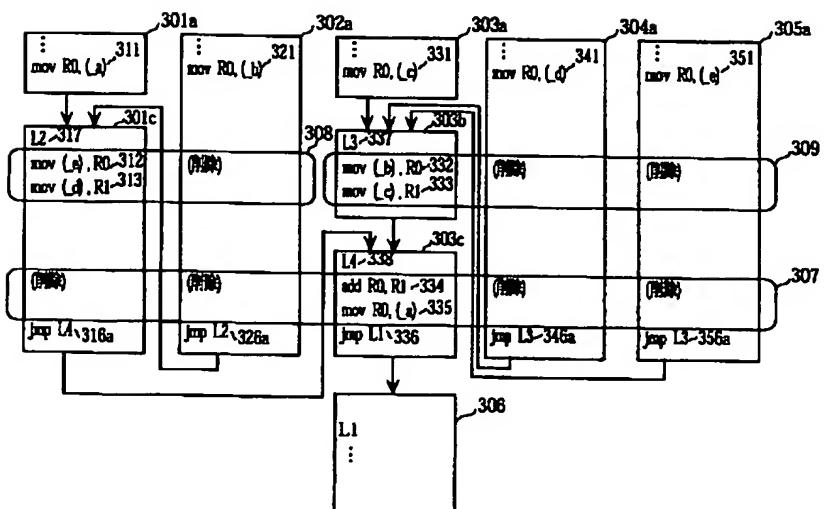
【図11】



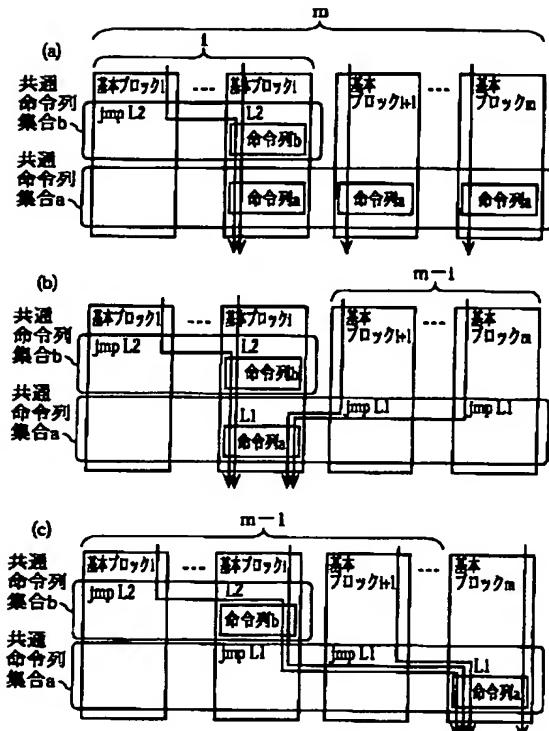
【図13】



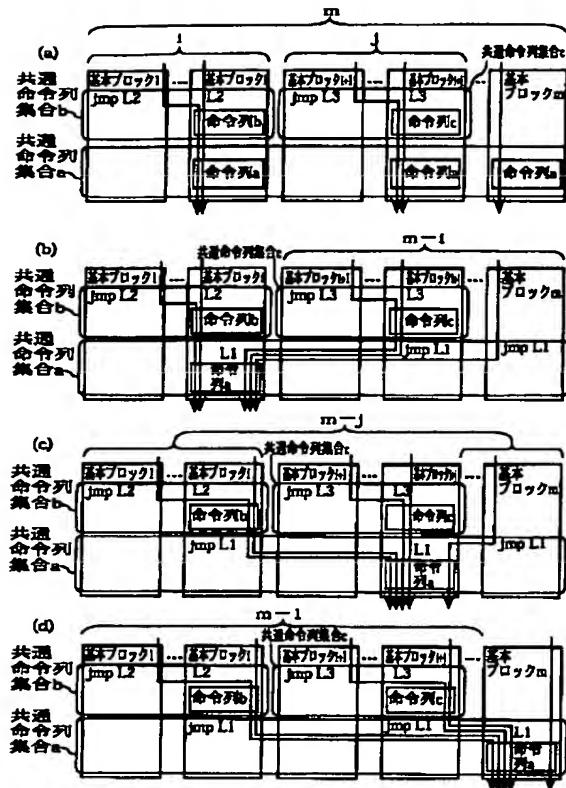
【図12】



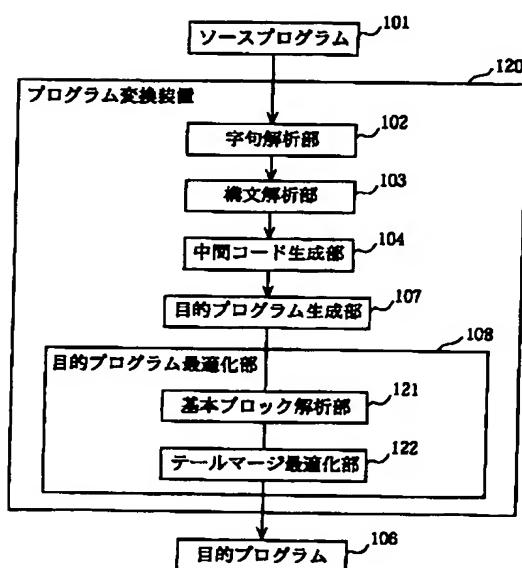
【図14】



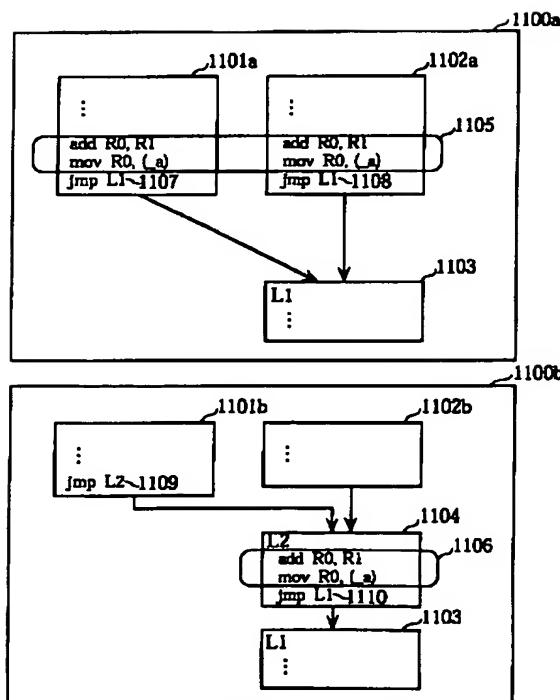
【図15】



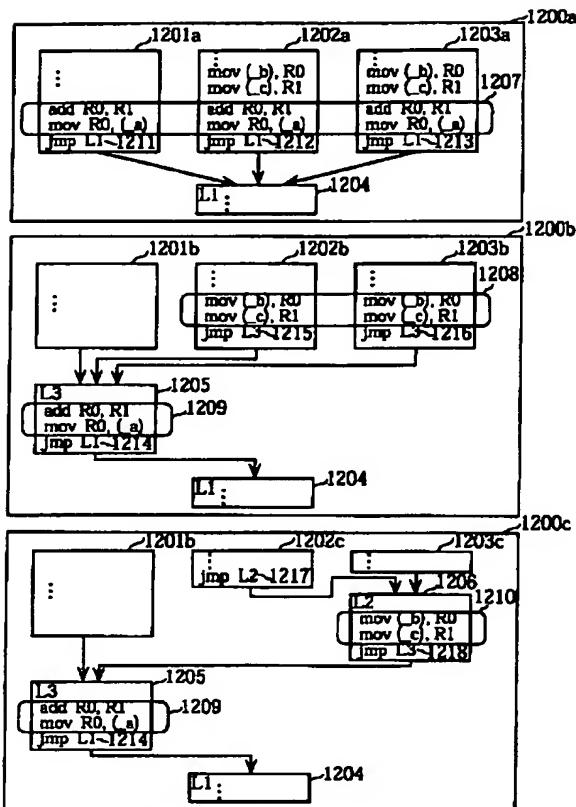
【図17】



【図19】



【図20】



【図21】

